Amendment "B"

Amendments to the Claims:

Claims 21-40 are added. Please amend claims 1, 9 and 17 as indicated below. The state of the claims following this Amendment "B" is as follows:

Claim 1 (currently amended). A method for automated testing of a graphical user interface (GUI) of a program, said method comprising:

creating a test case file comprising a plurality of test steps in a text format wherein the test steps are not written in an interpreted computer programming language; and

executing a test harness with said test case file as input to said test harness, said test harness configured to execute one of a plurality of automated tests in response to one of a plurality of test steps, each automated test configured to test a corresponding user interface element of said program through a GUI map, said GUI map configured to define a logical name for each user interface element of said program.

Claim 2 (original). The method for automated testing of a GUI of a program according to claim 1, wherein each test step comprises an object, an action, and an identification reference.

Claim 3 (original). The method for automated testing of a GUI of a program according to wherein each test step further comprises an optional field value.

Claim 4 (original). The method for automated testing of a GUI of a program according to claim 3, wherein each test step further comprises an error recovery value.

Claim 5 (original). The method for automated testing of a GUI of a program according to claim 1, further comprising:

generating said GUI map of said program by extracting a logical name, a physical name, an identification, and an ordinal value for each user interface element of said program.

Claim 6 (original). The method for automated testing of a GUI of a program according to claim 1, further comprising:

generating said GUI map of said program from one of a prototype of said program, a design document of said program and an earlier version of said program.

Claim 7 (original). The method for automated testing of a GUI of a program according to claim 1, wherein:

each automated test is further configured to retrieve and to execute at least one of a plurality of associated reusable functions in response to said one of said plurality of test steps.

Claim 8 (original). The method for automated testing of a GUI of a program according to claim 1, further comprising:

outputting results of the execution of said plurality of automated tests in response to said test file.

Claim 9 (currently amended). A system for automated testing of a graphical user interface (GUI) of an application, said system comprising:

at least one processor;

a memory coupled to said at least one processor;

a test harness residing in said memory and executed by said at least one processor, wherein said test harness is configured to execute one of a plurality of automated tests in response to one of a plurality of test steps of a text format test data file wherein the test steps of the text format test data file are not written in an interpreted computer programming language, each automated test configured to test a corresponding user interface element of said application through a GUI map, said GUI map configured to define a logical name for each user interface element of said application.

Claim 10 (original). The system for automated testing of a GUI of an application according to claim 9, wherein each test step comprises an object, an action, and an identification reference.

Claim 11 (original). The system for automated testing of a GUI of an application according to claim 10, wherein each test step further comprises an optional field value.

Claim 12 (original). The system for automated testing of a GUI of an application according to claim 11, wherein each test step further comprises an error recovery value.

Claim 13 (original). The system for automated testing of a GUI of an application according to claim 9, wherein said GUI map of said application is generated with a GUI analyzer configured to extract a logical name, a physical name, an identification and an ordinal value for each user interface element of said application.

Claim 14 (original). The system for automated testing of a GUI of an application according to claim 9, wherein said GUI map of said application is generated from one of a prototype of said application, a design document of said application, and an earlier version of said application.

Claim 15 (original). The system for automated testing of a GUI of an application according to claim 9, wherein each automated test is further configured to retrieve and to execute at least one of a plurality of associated reusable functions in response to said one of said plurality of test steps.

Claim 16 (original). The system for automated testing of a GUI of an application according to claim 9, wherein said test harness is further configured to generate an output file configured to contain results of said execution of said plurality of automated tests in response to said test file.

(Continued on next page.)

Claim 17 (currently amended). A computer readable storage medium on which is embedded one or more computer programs, said one or more computer programs implementing a method for automated testing of a graphical user interface (GUI) of an application, said one or more computer programs comprising a set of instructions for:

creating a test case file comprising a plurality of test steps in a text format wherein the test steps are not written in an interpreted computer programming language; and

executing a test harness with said test case file as input to said test harness, said test harness configured to execute one of a plurality of automated tests in response to one of a plurality of test steps, each automated test configured to test a corresponding user interface element of said program through a GUI map, said GUI map configured to define a logical name for each user interface element of said program.

Claim 18 (original). The computer readable storage medium in according to claim 17, said one or more computer programs further comprising a set of instructions for:

generating said GUI map of said program by extracting a logical name, a physical name, an identification, and an ordinal value for each physical element of said program.

Claim 19 (original). The computer readable storage medium in according to claim 17, said one or more computer programs further comprising a set of instructions for:

outputting an output file configured to contain results of the execution of said plurality of automated tests in response to said test file.

Claim 20 (original). The computer readable storage medium in according to claim 17, wherein said one or more computer programs further comprising a set of instructions for:

each automated test further configured to retrieve and to execute at least one of a plurality of associated reusable functions in response to said one of said plurality of test steps.

(Continued on next page.)

Claim 21 (new). A method for automated testing of a graphical user interface of a program, comprising:

receiving a test case file to a test harness wherein the test case file is in a text format and wherein the test case file is not written in an interpreted programming language;

reading the test case file;

determining a test step based on the test case file wherein each line of the test case file is interpreted by the software harness as a step of a test of the graphical user interface of the program; and

invoking an automated test routine from the test harness responsive to the test step determination wherein each automated test routine is configured to test a corresponding physical user interface element through a GUI map of the program.

Claim 22 (new). The method of claim 21, wherein each test step comprises an object, an action, and an identification reference.

Claim 23 (new). The method of claim 21, wherein each test step further comprises a field value.

Claim 24 (new). The method of claim 23, wherein each test step further comprises an error recovery value.

Claim 25 (new). The method of claim 21, further comprising:

generating the GUI map of the program by extracting a logical name, a physical name, an identification, and an ordinal value for each user interface element of the program.

Claim 26 (original). The method of claim 21, further comprising:

generating the GUI map of the program from one of a prototype of the program, a design document of the program and an earlier version of the program.

Claim 27 (original). The method of claim 21, wherein each automated test routine is further configured to retrieve and to execute at least one of a plurality of associated reusable functions in response to the test step.

Claim 28 (new). The method of claim 21, further comprising generating an output file of the results of the execution of the automated test.

Claim 29 (new). A system for automated testing of a graphical user interface (GUI) of a program, comprising:

at least one processing means;

computer readable memory means which is readable by the processing means, the computer readable memory means containing a test harness program comprising a series of computer executable steps configured to cause the processing means to:

control the reception of a test case file to a test harness wherein the test case file is in a text format and wherein the test case file is not written in an interpreted programming language;

read the test case file;

determine a test step based on the test case file wherein each line of the test case file is interpreted by the software harness as a step of a test of the graphical user interface of the program; and

invoke an automated test routine from the test harness responsive to the test step determination wherein each automated test routine is configured to test a corresponding physical user interface element through a GUI map of the program.

Claim 30 (new). The system of claim 29, wherein each test step comprises an object, an action, and an identification reference.

Claim 31 (new). The system of claim 30, wherein each test step further comprises a field value.

Claim 32 (new). The system of claim 31, wherein each test step further comprises an error recovery value.

Claim 33 (new). The system of claim 29, wherein the GUI map of the program is generated with a GUI analyzer configured to extract a logical name, a physical name, an identification and an ordinal value for each user interface element of the program.

Claim 34 (new). The of claim 29, wherein the GUI map of the program is generated from one of a prototype of the program, a design document of the program, and an earlier version of the program.

Claim 35 (new). The system of claim 29, wherein each automated test routine is further configured to retrieve and to execute at least one of a plurality of associated reusable functions in response to the test step.

Claim 36 (new). The system of claim 29, wherein the test harness is further configured to generate an output file of the results of the execution of the test case file.

Claim 37 (new). A computer readable medium storing a computer program to implement automated testing of a graphical user interface (GUI) of a program, comprising:

computer readable code to control the reception of a test case file to a test harness wherein the test case file is in a text format and wherein the test case file is not written in an interpreted programming language;

computer readable code to read the test case file;

computer readable code to determine a test step based on the test case file wherein each line of the test case file is interpreted by the software harness as a step of a test of the graphical user interface of the program; and

computer readable code to invoke an automated test routine from the test harness responsive to the test step determination wherein each automated test routine is configured to test a corresponding physical user interface element through a GUI map of the program.

Claim 38 (new). The computer readable medium of claim 37, further comprising computer readable code to generate said GUI map of said program by extracting a logical name, a physical name, an identification, and an ordinal value for each physical element of said program.

Claim 39 (new). The computer readable medium of claim 37, further comprising computer readable code to generate an output file of the results of the execution of the test case file.

Claim 40 (new). The computer readable code of claim 37, wherein each automated test routine includes computer readable code further configured to retrieve and to execute at least one of a plurality of associated reusable functions in response to said one of said plurality of test steps.

(End of Amendment "B".)